IMPROVED ASYMMETRIC DATA TRAFFIC
THROUGHPUT IN CSMA/CA NETWORKS

Related Application

[0001]   This application is a conversion of Provisional Application No.

5   60/195,638 filed April 7, 2000.

Background of the Invention

[0002]   The present invention relates to wireless communication systems.

[0003]   A popular Medium Access Control (MAC) method for wireless local

area networks (WLAN) devices is the Carrier Sense Multiple-Access/Collision

10   Avoidance (CSMA/CA). This method works by measuring the time since the last

activity on the shared channel in equal length contention slots. Each transmitting

node in the network generates a random number, then counts the number of

contention slots until the number is reached. At that point, the node can grab the

channel and other nodes must suspend their count until the channel is free again. If,

15   by chance, two nodes generate the same random number and, thus, collide, they

determine that the packet transfer is unsuccessful through the acknowledgment

process and generate a new random number to start counting slots from zero the

next time the channel is free.

[0004]   After a node has successfully received a message, it sends an
acknowledgment back over the wireless communication channel to the transmitting
node. The other nodes do not begin contending for the medium until after the
acknowledgment is sent by the receiving node. Note that typically, for local area
networks, above the MAC layer are higher level communication layers, such as
TCP/IP layers, under the control of the operating system. These TCP\IP layers have
their own message and acknowledgment signals. For Windows-based systems,
typically the TCP protocol is fixed to have a single acknowledgment for every two
TCP message transfers. The acknowledgment at the TCP level is distinct from the
acknowledgment at the MAC layer. In fact, each TCP acknowledgment requires a
MAC layer message containing the acknowledgment with its own MAC layer
acknowledgment of the TCP acknowledgment message.

[0005]   A key metric for any wireless local area network system is the peak data
rate in transferring data from one node to another without no other intentional traffic
on the network. For most applications today, the data transfer will occur using the
Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP decouples the
user from the details of the physical media and MAC layer use for the data transfer.
However, in the case of wireless LANs, the interaction between the lower layer
Medium Access Control (MAC) protocol and the upper layer (TCP) protocol has a
serious impact on the data rate especially in the peak scenario. In a typical TCP
implementation, such as those used in Windows based systems, an acknowledgment
is sent for every other TCP segment of 1460-bytes. Thus, in a system where, for
example, two TCP segments are sent in every CSMA/CA data packet, another tiny
64-byte payload packet TCP acknowledgment will contend for the media in
response to every MAC layer message even though the only real data traffic is one-
way which should not cause contentions. The effect of the small contending TCP
acknowledgment packets on data throughput can be severe. Even though the
message payload is small (only 64 bytes), all the overhead is required as for a large
packet. For example, in the HomeRF Shared Wireless Access Protocol (SWAP)

specification at 10 Mb/s data payload rate, the 1460-byte TCP segment message packet is typically about 3 ms long, while a 64-byte contending acknowledgment packet is about 2.4 ms long. If sent for every other message payload, this alone would be a 40% decrease in throughput. Additionally, CSMA/CA collisions can occur when both the TCP packet and the TCP acknowledgment packet randomly choose the same slot number. Dead time is created from the collision and both contending packets can see additional delays due to exponentially increasing contention times.

Brief Description of the Drawings

[0006]   Fig. 1A illustrates a prior art example of a wireless MAC layer signaling.

[0007]   Fig. 1B illustrates a modified wireless MAC layer signaling of the method of the present invention.

[0008]   Fig. 2A illustrates a prior-art communication between two nodes along a wireless medium.

[0009]   Fig. 2B illustrates the improved signaling along the wireless medium for an embodiment of the present invention.

[0010]   Fig. 3 is a flowchart that illustrates a method of one embodiment of the present invention.

Brief Description of the Preferred Embodiment

[0011]   Fig. 1 illustrates prior-art MAC layer signaling, for example, the signaling used in the CSMA/CA. In this embodiment, after a contention period, node A has access to the medium and transmits a data message 20 to the node B. The node B, after receiving the data message 20, sends a MAC layer acknowledgment 22 to the node A. Since all of the nodes expect the MAC layer acknowledgment 22 after the data message, the contention period for the nodes occurs only after the expected time for the MAC acknowledgment 22.

[0012]   Fig. 1B illustrates the method of the present invention. In this method, in some cases, after a data message 24 is transmitted from node A to node B, node sends back both a MAC acknowledgment 26 and an appended TCP acknowledgment 28. This TCP acknowledgment typically has about 64 bytes of data and, thus, can be easily appended to the MAC acknowledgment without significantly slowing the operation of the system. The other nodes in the system will wait until the end of the TCP acknowledgment before contending for the medium. Note that the system is backward compatible with prior versions of the wireless MAC protocols since the prior wireless MAC protocols need not know that the TCP acknowledgment 28 is appended to the MAC acknowledgment 26 in order not to interfere with the transmissions from node B.

[0013]   Fig. 2A illustrates a prior-art transmission across the wireless medium. A message 30 is transferred from wireless MAC node A to wireless MAC node B. A MAC acknowledgment 32 is then automatically transferred from node B to node A. After the TCP node B receives a number of messages, a TCP ACK message is produced to respond. Typically the TCP ACK message has 64-bytes of data, but will typically require a lot of overhead. The wireless node B needs to transfer the TCP acknowledge to the node A. This requires node B to contend for the medium even if node B is receiving data asymmetrically. For example, if node B is receiving a large amount of data, such as from a web page, from node A, the TCP acknowledgment still needs to transmitted as a MAC layer signal shown in Step 34. This MAC layer message containing the TCP ACK has its own MAC acknowledgment 36 from node A. Note that even though the TCP acknowledgment message is only 64 bytes long, it can use a lot of the wireless medium bandwidth, due to both the transmission overhead as well as the contention period.

[0014]   In the contention period, the nodes randomly choose a slot to begin the transmission. If a relatively small number of slots are used, the number of potential conflicts of the transmissions can be high. A small number of slots would otherwise be useful in a situation where the asymmetric data transfer is common. However,

the need to transmit TCP acknowledgments reduces the desirability of using a small number of slots. A problem with using a relatively large number of slots is that it increases the system delay before a node can transmit.

[0015]   In the embodiment of Fig. 2B, once a TCP acknowledgment signal is prepared by the TCP node B, the TCP acknowledgment signal is buffered and then appended to the MAC acknowledge signal in Step 40. The TCP acknowledgment responses from node B to node A are appended to the relatively short MAC acknowledgment, which need not contend for the medium. In one embodiment, the MAC layer acknowledgment of the TCP ACK is appended to another message 42 from node A to node B. The advantage of the system of Fig. 2B is that the overhead associated with the TCP acknowledge signal is significantly reduced, as well as contention avoided. As long as node A is sending data to node B, node B need not contend for the wireless medium.

[0016]   In another embodiment, the TCP ACK message is appended only if link quality is excellent and hence MAC acknowledgments of the TCP ACK message is not required. In the rare occurrence that this leads to message delivery failure, the TCP layer will detect then failure and retransmit as required.

[0017]   Fig. 3 is a flowchart that illustrates an asymmetric receive mode for the wireless MAC layer. In Block 70, TCP acknowledgment is ready. In Block 72, checking is done to see whether the system is in the asymmetrical receive mode. If not, the system contends to transmit the TCP acknowledgment across the wireless medium in Step 74. If the system is in asymmetrical receive mode, the system buffers the TCP acknowledgment. The system stores the TCP acknowledgment until a MAC acknowledgment is ready to send. In Step 78, it is checked to see whether a timer period is over. If this time period is over, rather than continue to wait, the system moves to Step 74 to contend to transmit the TCP acknowledgment. In this way, the system does not buffer the TCP acknowledgment indefinitely.

[0018]   In Step 80, it is checked to see whether an additional MAC layer acknowledgment is available to send. If not, the system loops back and waits. If so,

however, in Step 82, the system transmits the TCP acknowledgment appended to the MAC layer acknowledgment.

[0019]   Appending the TCP acknowledgment to the MAC layer ACK has two major impacts which improve CSMA/CA peak data rates:

[0020]   1. The incremental time for the TCP ACK is simply the 64-byte length (plus an acknowledgment bit in the header of subsequent A → B packets). Thus the 2.4 ms TCP ACK packet of the previous SWAP example is now only about 0.1 ms.

[0021]   2. The TCP ACK short messages are now completely removed from contention. This allows the system to optimize smaller contention windows (since no collisions occur) and longer multi-segment packets (since no collisions occur).

[0022]   For the case of the Home RF SWAP protocol using a 10 Mb/s payload rate, the increase in peak data rate from this one new MAC technique is considerable - approximately 3 Mb/s to 5.5 Mb/s effective TCP throughput. The exact percentage improvement may be different for other WLAN protocols using CSMA/CA but it will always be positive for peak data rate.

[0023]   The effect is diminished if the TCP Receive ACK Count is increased to much higher numbers such as 8 or 16 (Note: TCP Window size must also increase accordingly). However, there are currently nearly 100 million Windows-based devices for which ACK Count = 2 is fixed and this new MAC protocol for WLANs based on CSMA/CA will benefit these existing devices.

[0024]   In one embodiment, the nodes will buffer all messages to be sent that are below a predetermined size (these messages most likely being TCP ACKs). This buffering is preferably done when the channel quality is relatively good. The small sized messages are appended to a transmitter MAC acknowledgment.

[0025]   It will be appreciated by those of ordinary skill in the art that the invention can be implemented in other specific forms without departing from the spirit or character thereof. The presently disclosed embodiments are therefore

considered in all respects to be illustrative and not restrictive. The scope of the invention is illustrated by the appended claims rather than the foregoing description, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced herein.